

REMARKS

In response to the Office Action mailed May 24, 2004, Applicant respectfully requests reconsideration.

As a preliminary matter, Applicant notes with appreciation the indication of allowable subject matter in claims 9 and 10.

Claims 11-13 were rejected under 35 U.S.C. §102(e) as being anticipated by Eidt et al. Applicant respectfully traverses this rejection.

Eidt describes a loader program for loading a relocatable file into the memory of a computer for execution. This program uses an object code file (known as a PEF container) with certain loading instructions contained therein. These instructions allow the PEF container to be smaller than conventional files. This is achieved by using an “abstract machine” to interpret the relocation instructions, which allows certain parameters to be calculated or derived from the type of instruction, without explicitly stating their values in the relocation instruction, thereby saving the amount of data that needs to be stored. Furthermore, the number of instructions is reduced by defining a single instruction that allows the same relocation instruction to be performed on a run of consecutive items. This prevents the same instruction being repeated several times in the file.

By contract, claim 11 recites an assembler for assembling an object code module from a source code module comprising program instructions, data and assembler directives that are executable to generate relocation instructions in the object code module, the assembler comprising, *inter alia*, a source code reader for reading said program instructions and directives from the source code module, and a directive translator for generating compound relocation instructions. Eidt does not teach or suggest a source code reader, as Eidt takes as its input object code files that have already been compiled from source code. Furthermore, Eidt does not teach or suggest a directive translator for generating compound relocation instructions. The relocation instructions in Eidt have already been generated and are present in the object code module, and Eidt makes no mention of producing compound relocations from sequences of instructions. Accordingly, claim 11 patentably distinguishes over Eidt and is in allowable condition. Claims

12-13 depend from claim 11 and are allowable for at least the same reasons. Withdrawal of the rejection under 35 U.S.C. §102(e) is respectfully requested.

Claims 1-8 and 14 were rejected under 35 U.S.C. §103(a) as being unpatentable over Eidt in view of Hastings. Applicant respectfully traverses this rejection.

Eidt has been discussed above.

Hastings is directed to a method for inserting new instructions and data in an object code file, between existing instructions and data. The offsets are then modified in order to account for the new locations of the preexisting instructions and data. Although Hastings does disclose insertion into object code, the method of Hastings is concerned with inserting object code instructions between preexisting object code instructions in the section data of the object code module. This is not the same as replacing a sequence of relocation instructions with a single compound relocation instruction. In particular, claim 1 recites a method of assembling a source code module to form an object code module, said source code module including one or more assembler directives, wherein the assembler directives are used to generate relocation instructions in the object code module, the method comprising, *inter alia*, determining if said associated sequence of relocations matches one of said stored sequence of relocation instructions and if a match of relocation sequences is determined, inserting into said object code module a compound relocation including the compound relocation indicator of said matched compound relocation sequence definition instruction and said matched sequence. Accordingly, at least this limitation of claim 1 is not found in Hastings.

With respect to Eidt, claim 1 recites “reading a plurality of compound relocation sequence definitions stored in a memory, each compound relocation definition sequence comprising a compound relocation indicator and a first sequence of relocation instructions”. Clearly, Eidt does not teach or suggest anything that could be considered a “compound relocation” definition sequence or the step of reading such a sequence.

Claim 1 also recites “reading assembler source code from said source code module, said source code generating an associated sequence of relocation instructions for executing the directive”. Eidt clearly does not teach or suggest this limitation because Eidt is directed to a loader, not an assembler. Therefore, Eidt does not read source code, but object code that has

already been assembled. This can clearly be understood with reference to column 5, lines 56-59 of Eidt, where it is stated that the PEF container, which is the input to the loader, is formed from the compiled output of source code modules. Therefore, the relocation instructions in Eidt have already been generated and are present in the PEF container. Eidt makes no mention of how to generate these.

Accordingly, claim 1 clearly distinguishes over Eidt and Hastings individually or in any combination.

Claims 2-8 depend from claim 1 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 1-10 under 35 U.S.C. §103 is respectfully requested.

Claim 15 was rejected under 35 U.S.C. §103(a) as being unpatentable over Eidt in further view of the background section of Eidt. Applicant respectfully traverses this rejection.

Claim 15 recites, *inter alia*, a toolchain for forming a target executable program from a plurality of source code modules, each source code module comprising program instructions or data and assembler directives, the toolchain comprising, *inter alia*, an include source code module comprising a plurality of compound relocation definitions sequences, each compound relocation definition sequence comprising a compound relocation indicator and a first sequence of relocation instructions.

Contrary to the Office Action, no part of Eidt teaches or suggests the use of a “include source code module” in addition to a plurality of source code modules. Eidt also does not teach or suggest any source code module defining “compound definition sequences” as recited in claim 15. Accordingly, claim 15 clearly distinguishes over Eidt and is in allowable condition. Withdrawal of the rejection under 35 U.S.C. §103(a) of claim 15 is therefore respectfully requested.

Serial No.: 09/997,230
Conf. No.: 2536

-9 -

Art Unit: 2122

CONCLUSION

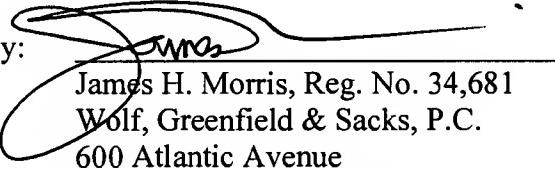
A Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Respectfully submitted,

Richard SHANN

By:


James H. Morris, Reg. No. 34,681

Wolf, Greenfield & Sacks, P.C.

600 Atlantic Avenue

Boston, Massachusetts 02210-2206

Telephone: (617) 646-8000

Docket No.: S1022.80806US00

Date: September 24, 2004

x9/24/04x